Lab 2: Digital Modulation & Transmission

EEP55C26 Open Reconfigurable Networks
Lingyu Gong

I. Lab purpose

The lab focuses on various forms of digital carrier modulation, evaluates bit error performance, and examines the effects of noise and interference. It also covers the importance of carrier and symbol synchronisation in digital communication systems, leveraging both theoretical models and practical application using software-defined radios (SDRs).

II. Digital carrier modulated waveforms

1. Creating the Simulink model

This part uses the same model as before, with a Centre Frequency and Pluto Receiver module. I added three filter designers to filter three different communication signals. Figure 1-1 shows my whole structure for part 1.

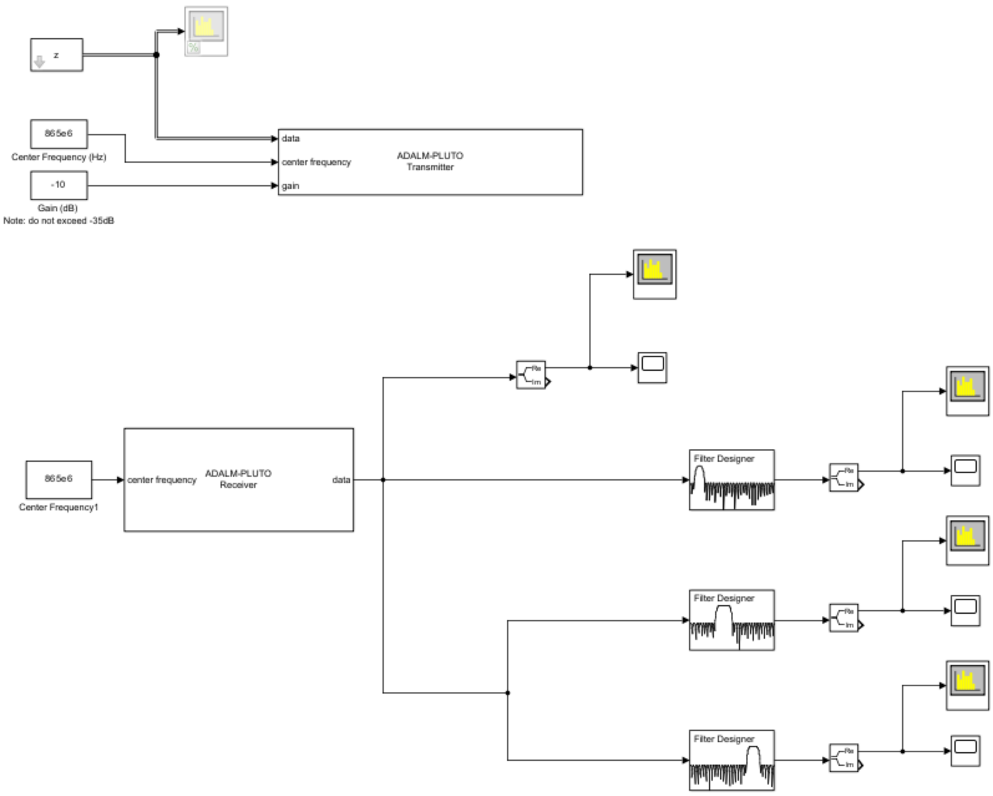

Figure 1-1 Simulink model for Part1.

I set the centre frequency to 865kHz and ran a model. The received signals are displayed in figure 1-2. From the graph, I observed that the signal frequencies were around 10kHz, 50kHz and 90kHz. Then, I created three filters and assessed their performance using a spectrum analyser. Figure 1-3 shows the result of the program in the time domain, but it is difficult to analyse without using filters to classify each signal and distinguish the differences clearly. This will help in performing a detailed analysis.

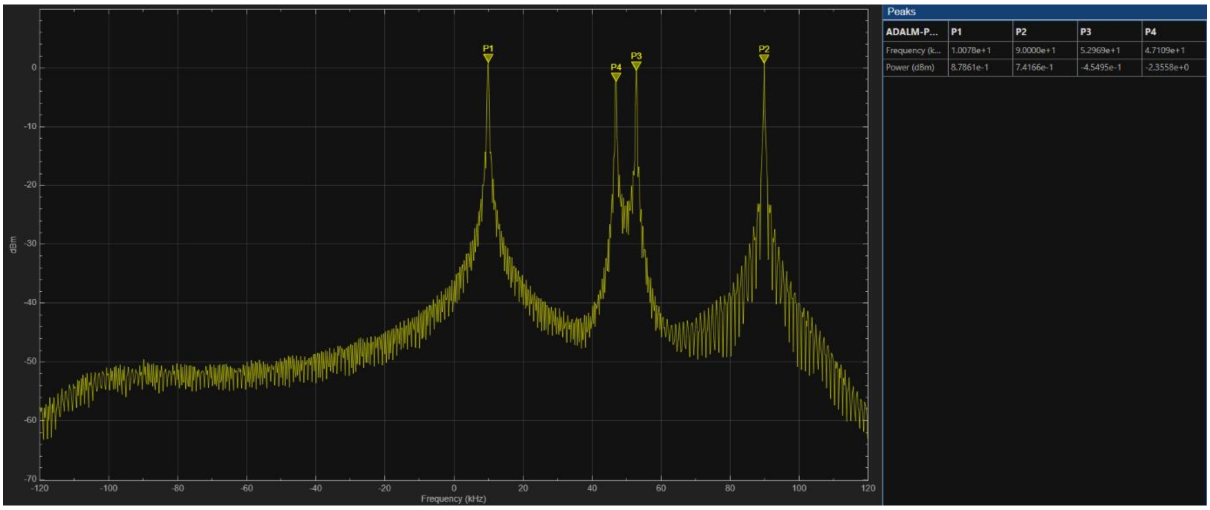

Figure 1-2 The signal captured by receiver.

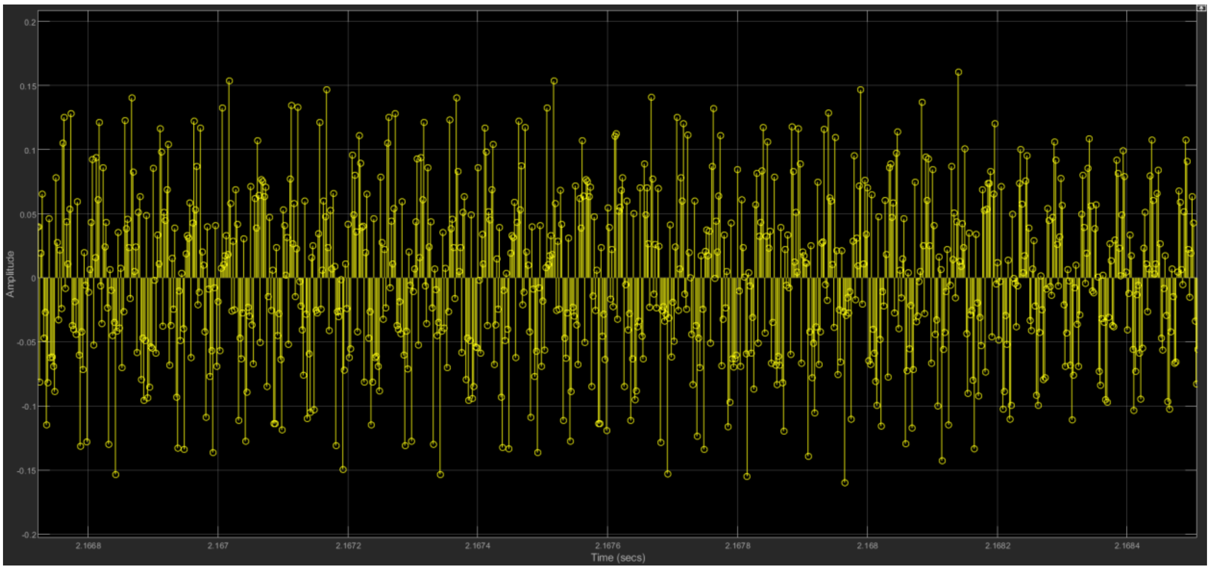

Figure 1-3 Time Domain Graph.

<u>Question 2.2: Identify each of the digital communication signals within your received signal. From the screen captures of Spectrum Analyzer and Time Scope, can you characterise each signal? Also, from the time/frequency plots, can you determine the bit rate for each signal and indicate how you determined them. Compare, and contrast the salient features seen in the plots.</u>

I start with a brief introduction about ASK, PSK and FSK, just to show my understanding. ASK (Amplitude Shift Keying) is a modulation technique that represents data by varying the amplitude of the carrier wave. It's simple but susceptible to noise. PSK (Phase Shift Keying) encodes data by changing the phase of the carrier wave. It is more robust against noise compared to ASK. FSK (Frequency Shift Keying) encodes information by varying the frequency of the carrier wave and is often more resistant to interference and easier to implement.

So next I will analyse three signals one by one. In the first graph (Figure 1-4), I show the Spectrum Analyzer of the first signal (around 10kHz) after a FIR filter. Figure 1-5 shows the time domain result for the signal after the FIR filter. Through observation, we can find that there is no great amplitude change in the overall view, and there is only a regular change in the middle, which is initially guessed to be a phase flip, so the first judgement is that the first signal uses the PSK method. I used red arrows to mark the position of the flip-flop and can guess that each change carries a class of information.

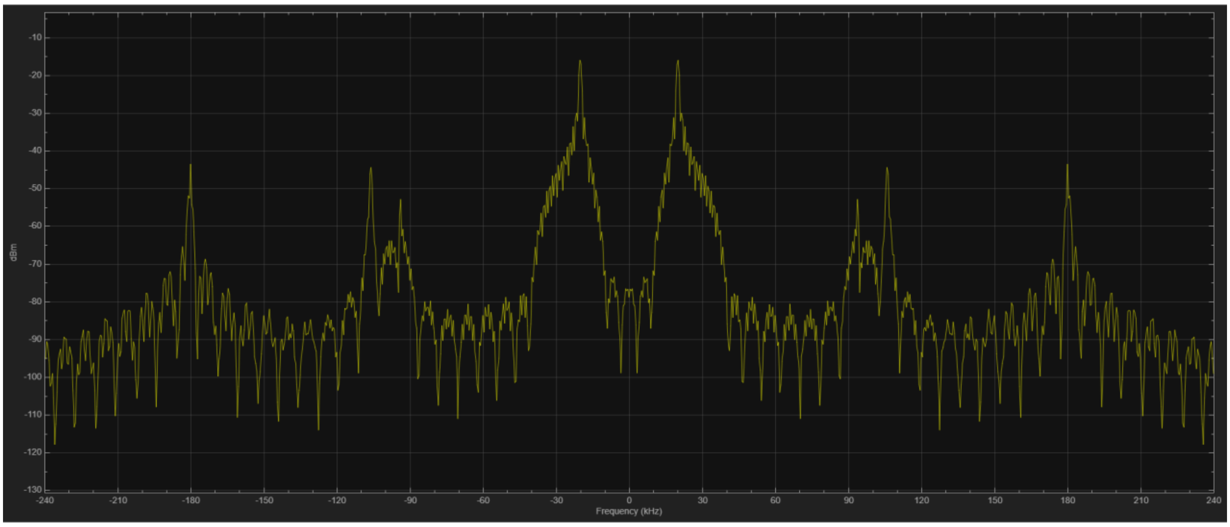

Figure 1-4 Spectrum Analyzer result for the first signal.

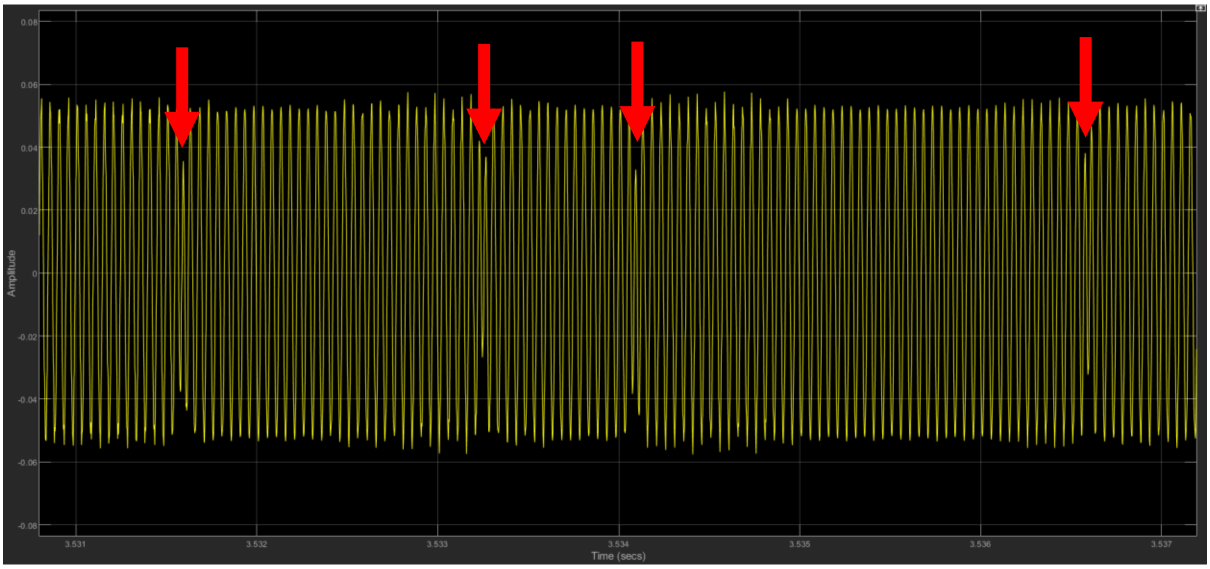

Figure 1-5 Time Domain result for the first signal.

I have used Fig. 1-6 and Fig. 1-7 to show the second signal obtained after the design of the filter designer, Fig. 6 shows the results from the spectrum analyser and Fig. 1-7 shows the graph obtained from the time domain analysis. The information that can be obtained from the time domain graph is that there is not much change in its amplitude and there are no significant jumps, but in combination with the presence of two peaks in the frequency domain, it can be guessed that it is in FSK mode.

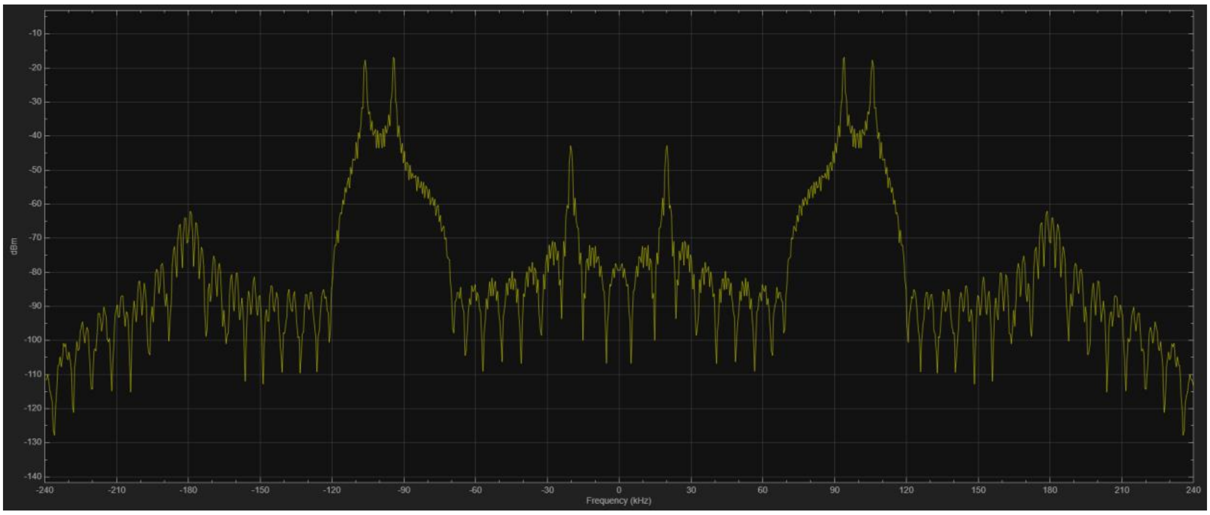

Figure 1-6 Spectrum Analyzer result for the second signal.

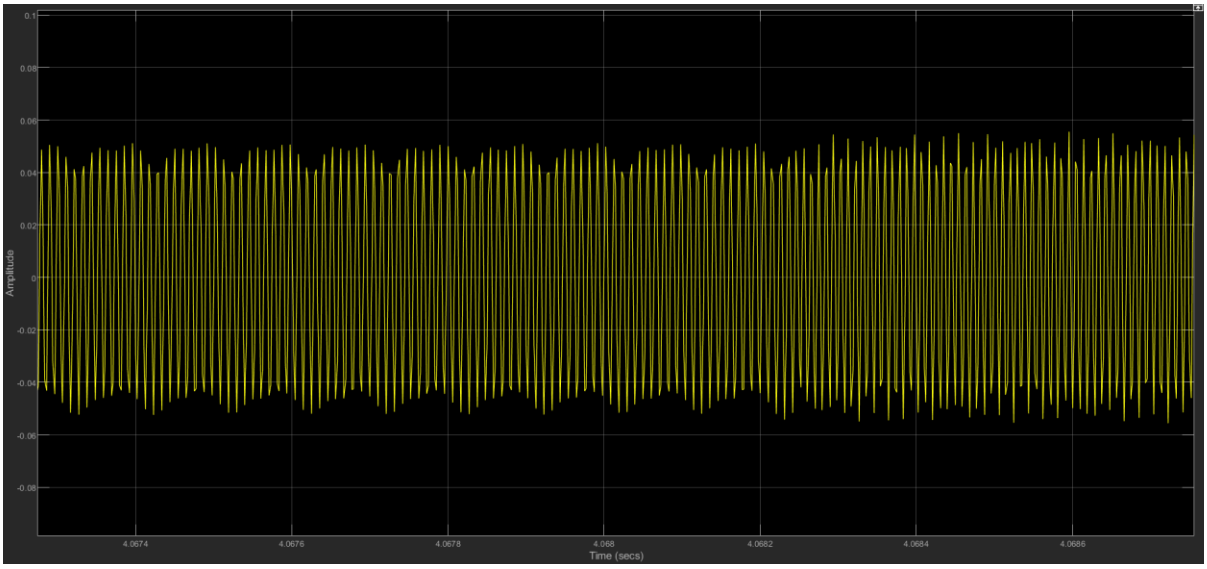

Figure 1-7 Time Domain result for the second signal.

Now it's time to analyse and judge the last signal waveform, I use Figure 1-8 and Figure 1-9 to show the Spectrum Analyzer result and Time Domain result respectively. from the Time Domain graph, you can clearly see that the amplitude is constantly changing (especially when it's changed to Stem display), so it's very consistent with ASK's characteristics, so I guess that the waveform is ASK signal.

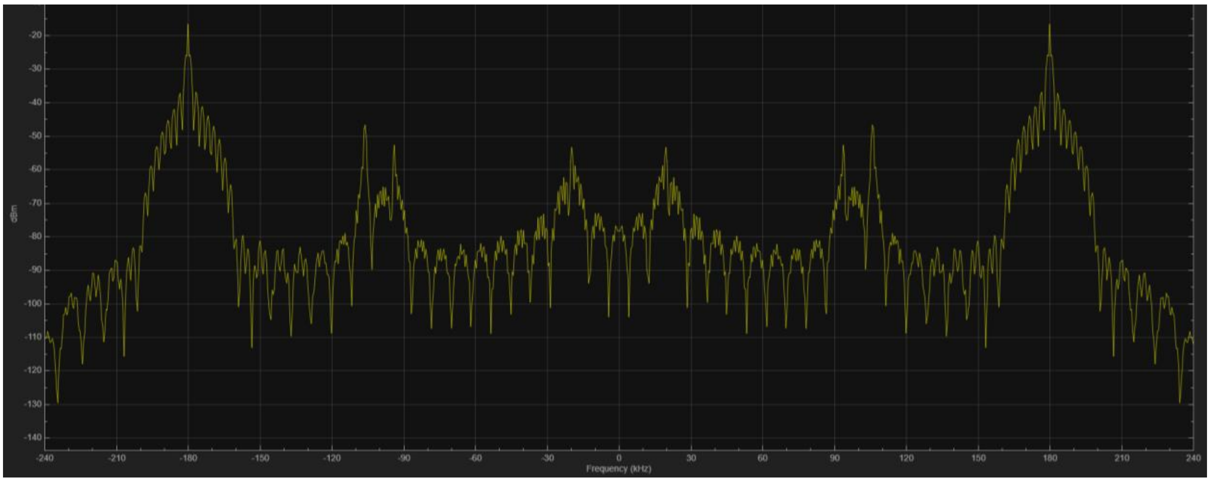

Figure 1-8 Spectrum Analyzer result for the third signal.

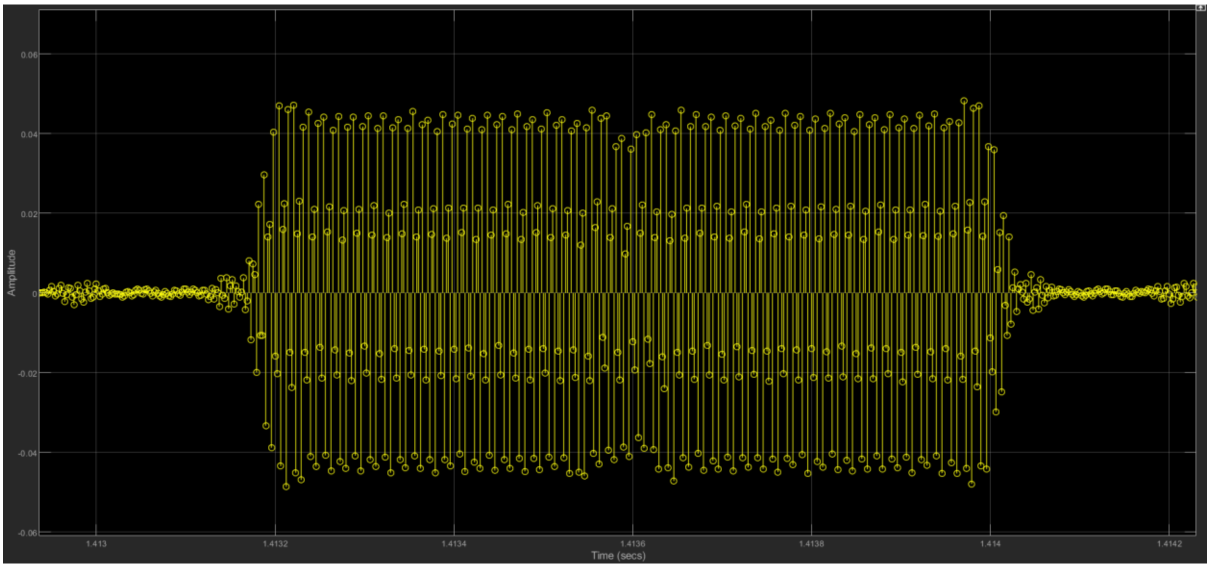

Figure 1-9 Time Domain result for the third signal.

Regarding the calculation of the bit rate, firstly the first one, which I judge to be a PSK signal, has a bit rate that is the reciprocal of this cycle multiplied by the number of bits per cycle, and I have intercepted one of the phase changes and know that this is a binary PSK, so calculations based on the data shown in Fig. 1-10 give a bit rate of approximately: 1166.774 bps.

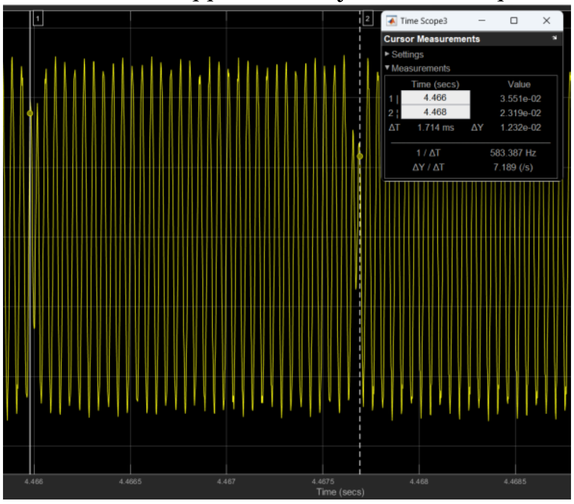

Figure 1-10 Bit rate calculation for PSK signals.

The same question as before, but this time I changed to the stem view to make it clearer. To calculate the bit rate for FSK signal, we need to measure the lengths of different frequency cycles in the time domain that correspond to the bits sent. The bit rate is obtained by taking the reciprocal of the cycle length. Based on the data in the figure 1-11, the conclusion is that the bit rate is approximately 2.000kHz.

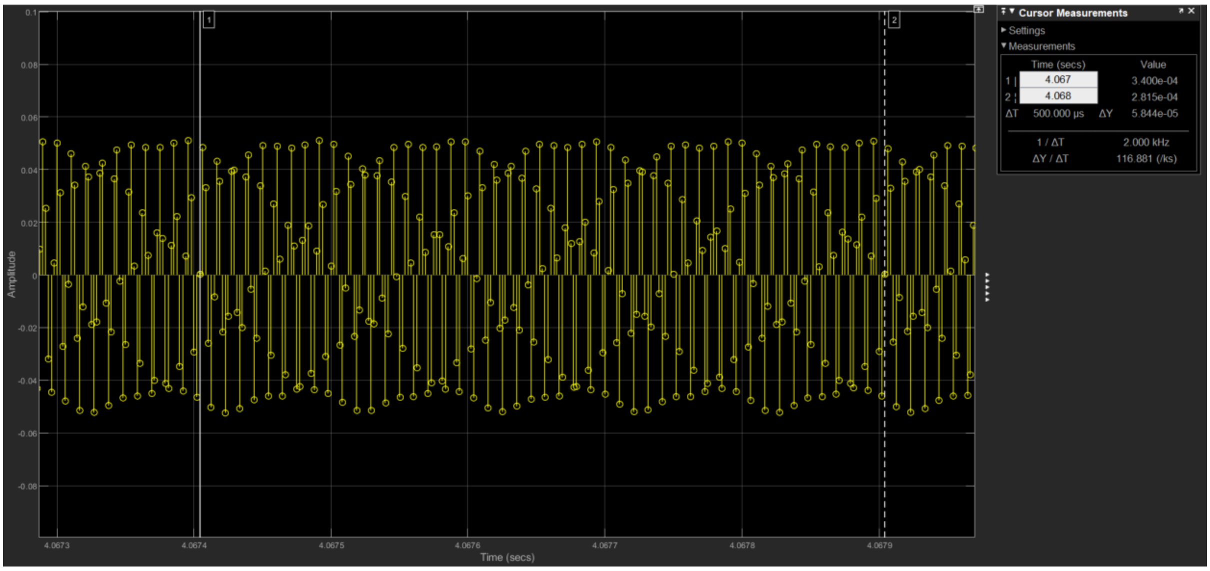

Figure 1-11 Bit rate calculation for FSK signals.

For ASK signals, if the change in each data bit is clearly visible in the time-domain waveform (e.g., the duration of a high level), then the bit rate is the reciprocal of that duration. The result shows on the graph (Figure 1-12): about 492.950kHz.

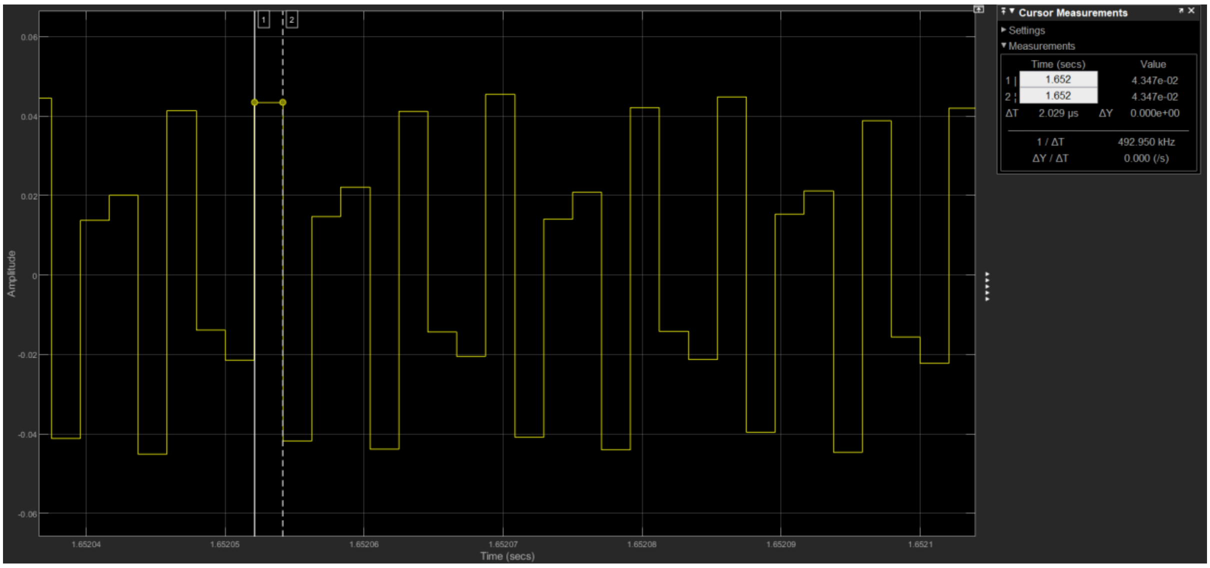

Figure 1-12 Bit rate calculation for FSK signals.

<u>Question 2.3: One of communication signals in your received wide-band signal use binary Amplitude Shift Keying (ASK), which is essentially a binarized amplitude modulated (DSBAM) waveform. Using your non-coherent DSBAM receiver from Lab 1, recover the actual baseband message. Add a screen capture of the Simulink model and a Time-Scope window showing the message waveform in your submission.</u>

As shown below in Figure 1-13, I used the same model in lab1 and increased the Gain to 50dB, then ran the code, I can see this signal change into a DSBAM signal, which has three peaks.

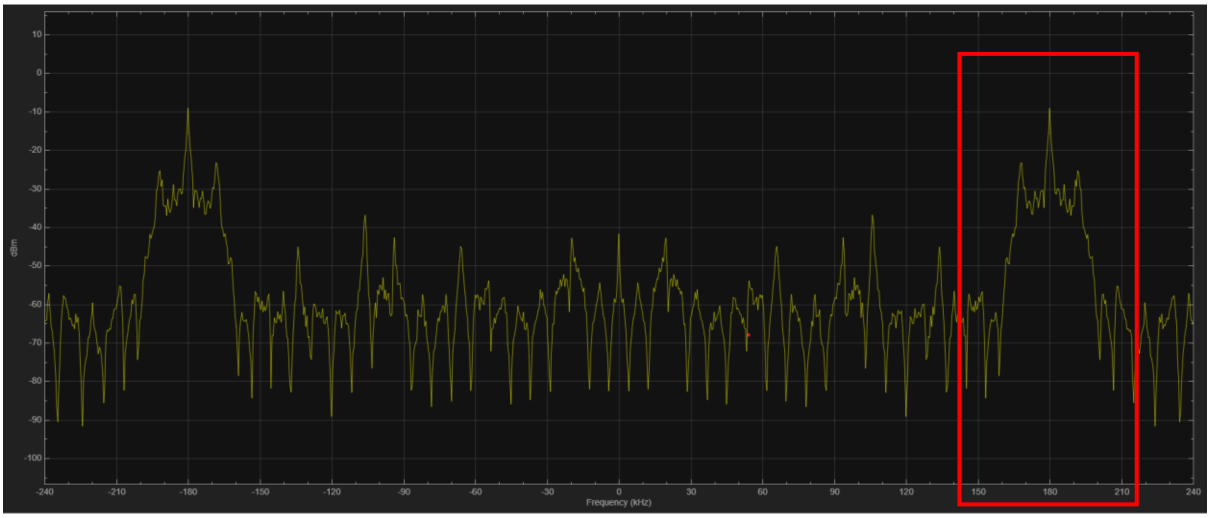

Figure 1-13 DSBAM signal after recovery.

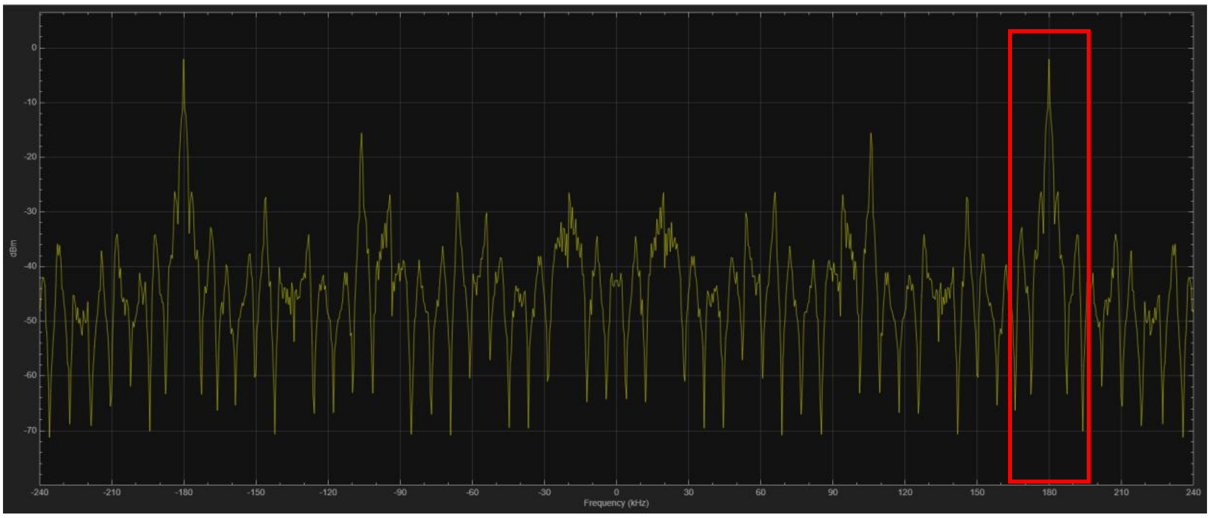

Figure 1-14 DSBAM recovery signal after filter in the frequency domain.

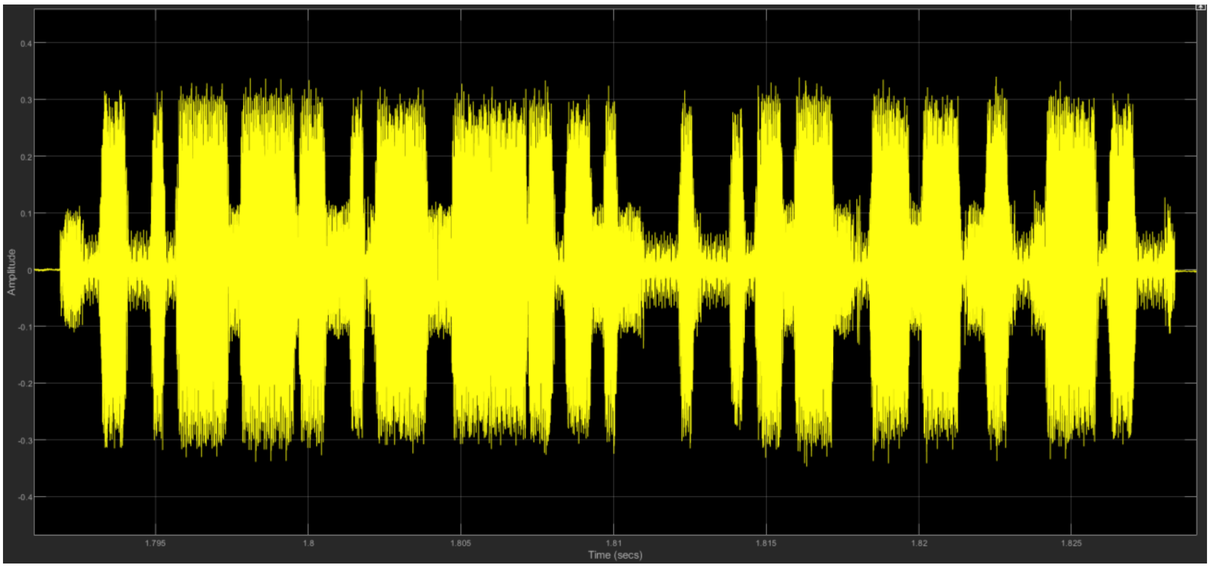

Figure 1-15 DSBAM recovery signal after filter in the time domain.

2. Bit Error Rate Simulation

Figure 2-1 shows the Simulink model to simulate BPSK communication.

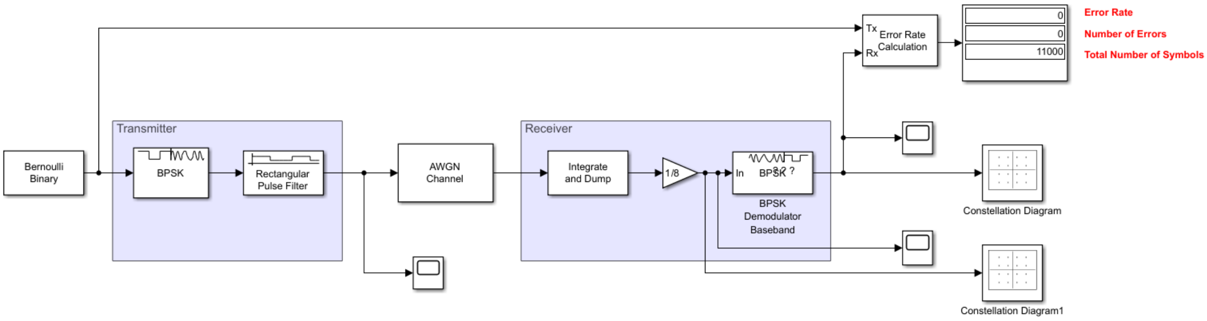

Figure 2-1 Simulink model to simulate BPSK communication

<u>Question 3.1: With a Time Scope, block added, use it to display the output of the Ideal Rectangular Pulse Filter block. Can you determine the average signal power?</u>

After incorporating a Time Scope subsequent to the Ideal Rectangular Pulse Filter block, we captured the signal's output as depicted in Figure 2-2. The figure illustrates the signal in a two-colour scheme, with each colour corresponding to the real and imaginary components of the signal, respectively. To compute the average power of the signal, we utilize the sum of the squares of the amplitude of the real part and the imaginary part. Referring to the data presented in Graph 2-3, we determine that the average signal power is approximately 0.99995 dB.

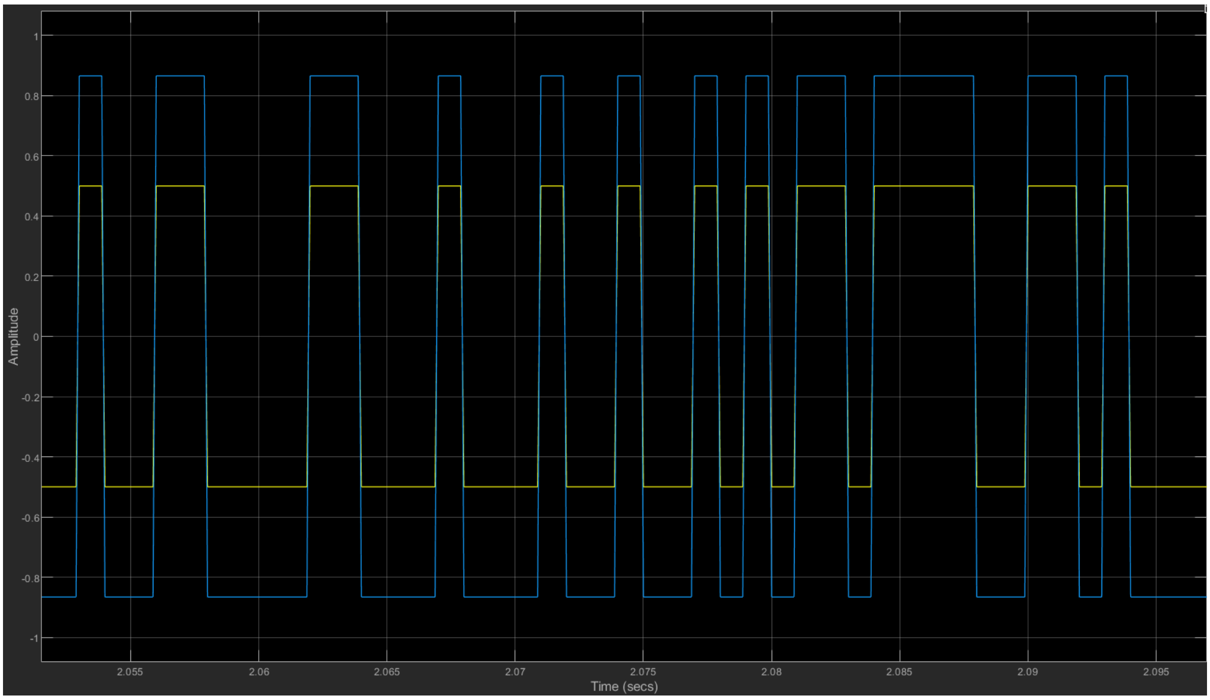

Figure 2-2 Ideal Rectangular Pulse Filter block on time domain

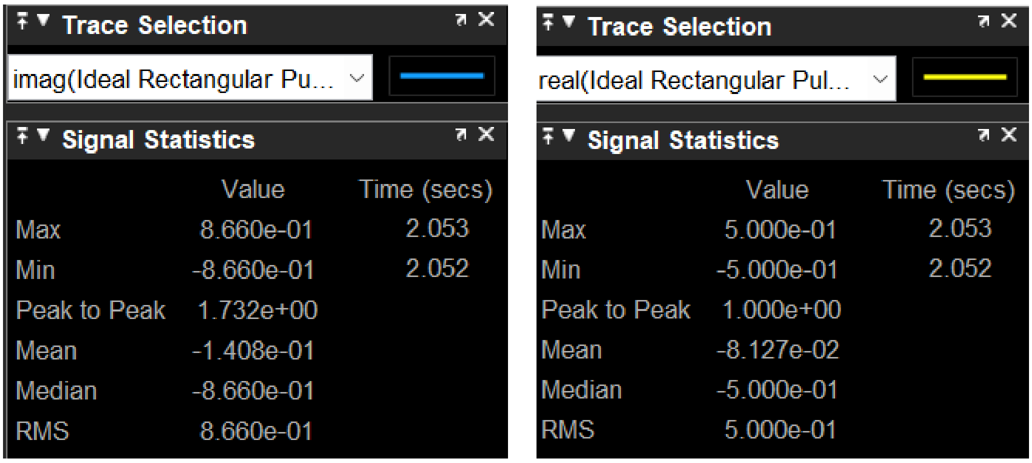

Figure 2-3 Real and Imag data

<u>**Question 3.2: Using a Time Scope block, observe the input to the AWGN Channel block, the output of the AWGN Channel block, and the output of the Gain block. In the AWGN Channel block, set the Input signal power to the value you calculated in the previous problem and set to Eb/N0 12 dB. Set the remaining parameters based on your understanding of the simulation. Run the simulation, and take a screen capture of the Time Scope block. Explain and interpret the plots.**</u>

I added two Time Scopes each at the front and back of the AWGN module, and another at the front and back of the Gain module, and set Eb/N0 to 12dB as requested.

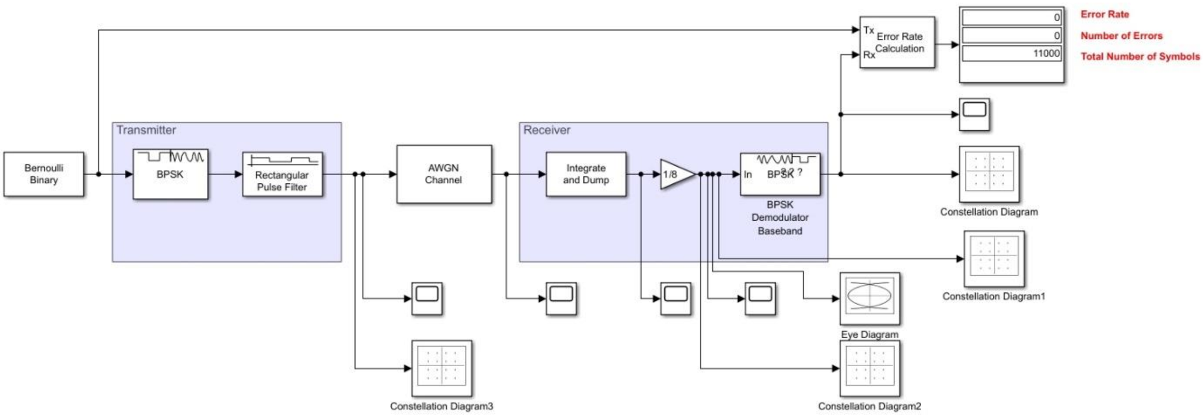

Figure 2-4 The structure of BPSK communication after modified.

After running the module simulation, the Time Scope before and after the AWGN module is expanded on the same page for comparison, and the effect of the added noise on the original signal can be clearly seen. Figure 2-5 The original signal at the top and the signal with added noise at the bottom.

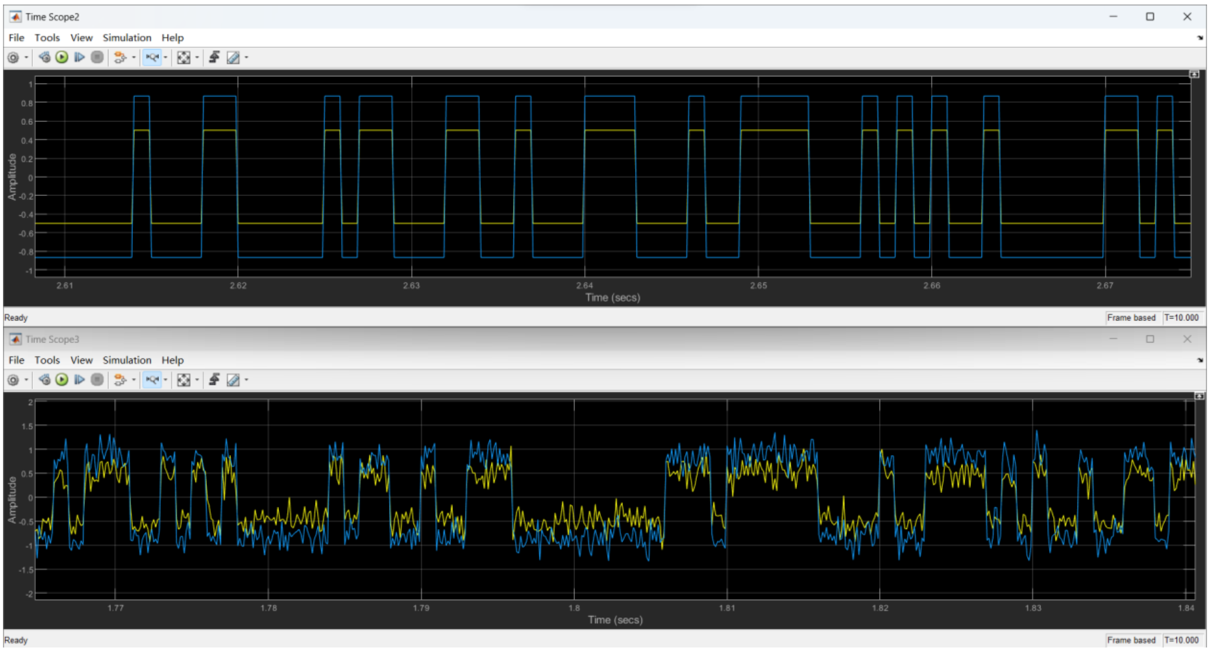

Figure 2-5 The top signal is the original signal and the bottom signal is the noise signal.

<u>Question 3.3: Add Eye Diagram and Constellation Diagram blocks to the signal at the input of the BPSK Demodulator Baseband block. Set the AWGN Channel block for Eb /No of 12 dB. Run the simulation, and take screen captures of the eye diagram and signal constellation. Repeat for Eb /No of 6dB. Comment on the observations and how the plots change as a function of Eb /No.</u>

Add Eye Diagram and Constellation Diagram blocks according to the title, first set the observation Eb /No to 12 dB for observation, then set this parameter to 6 dB, and put the two diagrams together for comparison, as shown in Figure 2-6.

Comparing Figures 2-6 and 2-7 it is clear that when the value of Eb /No is large, the eye shape formed is clearer and less noisy, but on the contrary when the value of Eb /No is small, there is a very large noise interference and a lot of noise.

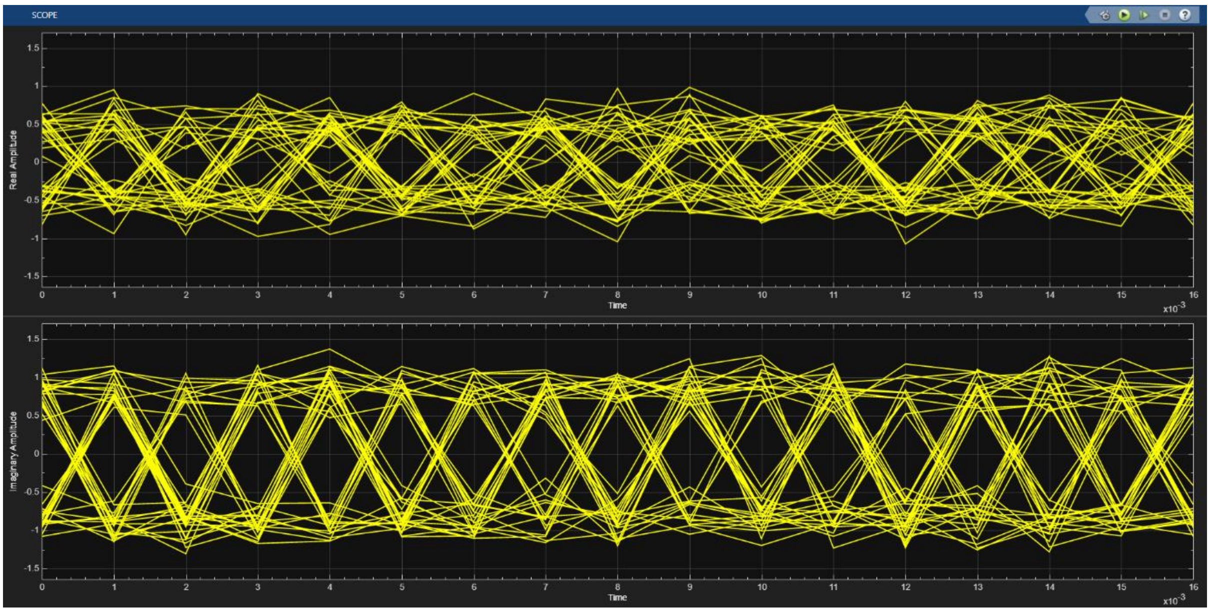

Figure 2-6 Eye Diagram with Eb/No=12dB.

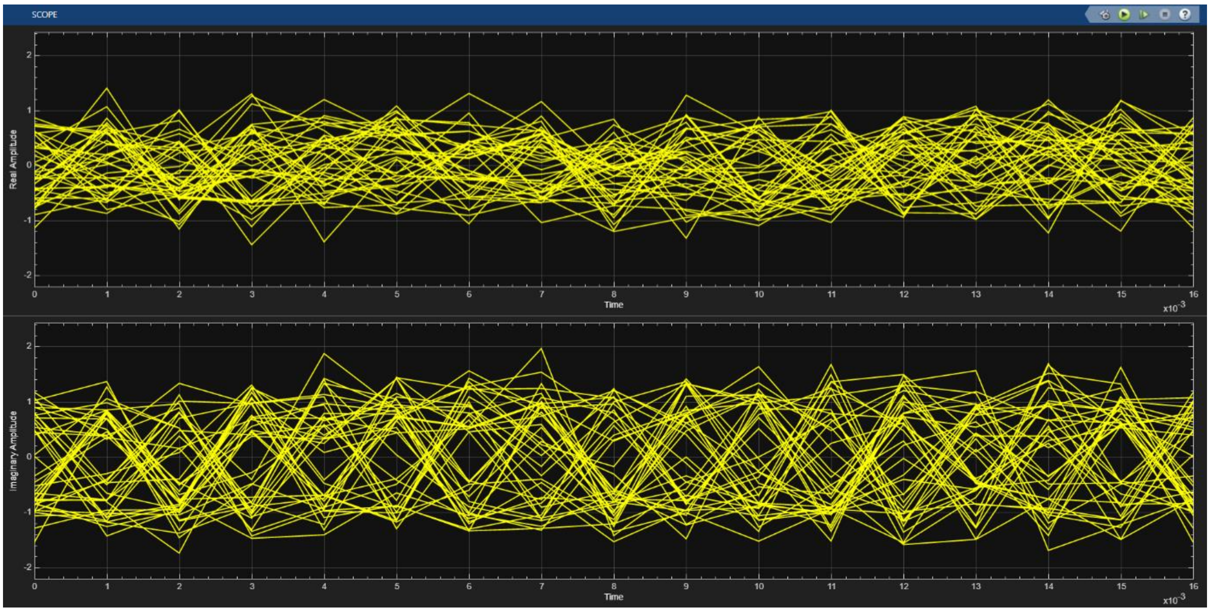

Figure 2-7 Eye Diagram with Eb/No=6dB.

Observing the constellation diagram, it is obvious that when the value of Eb/No is larger, the signals near the target point are more clustered, and vice versa, they become loose, which tends to bring about a larger error.

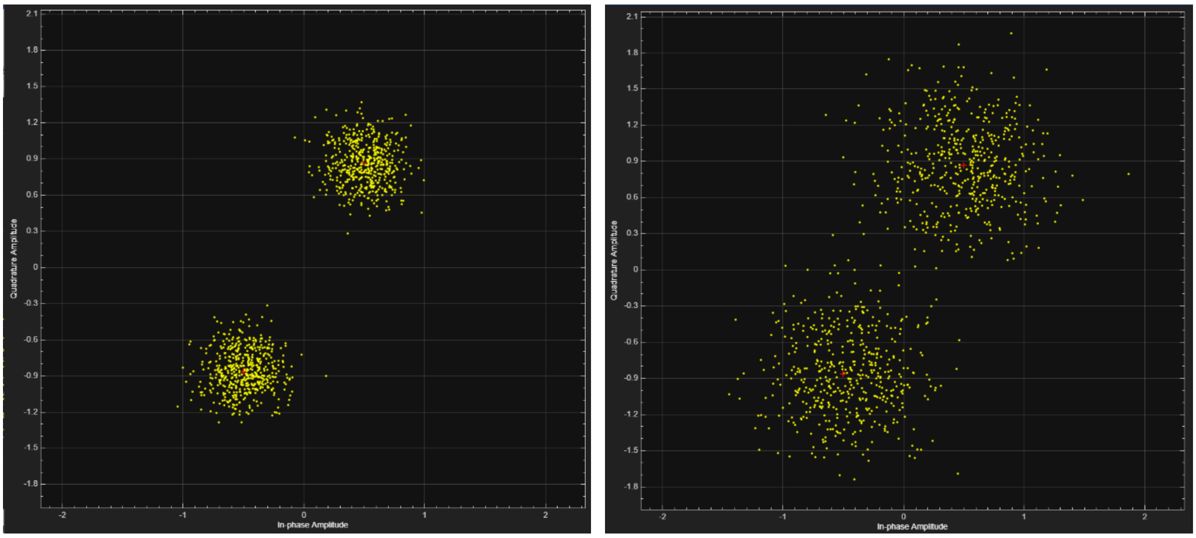

Figure 2-8 Comparison of Constellation Charts.

<u>Question 3.4: You will now use the model to simulate the system performance over a range of Eb /No values.</u>

<u>• Run the simulation for Eb /No values of 0dB to 12dB in 4dB steps.</u>
<u>• Set the simulation time to ensure the results are meaningful. This is especially important when very few bit errors are expected.</u>
<u>• Observe and record the error rate for each run.</u>
<u>• Plot bit error rate vs Eb /No and compare to the theoretical prediction. What are your thoughts on the results?</u>

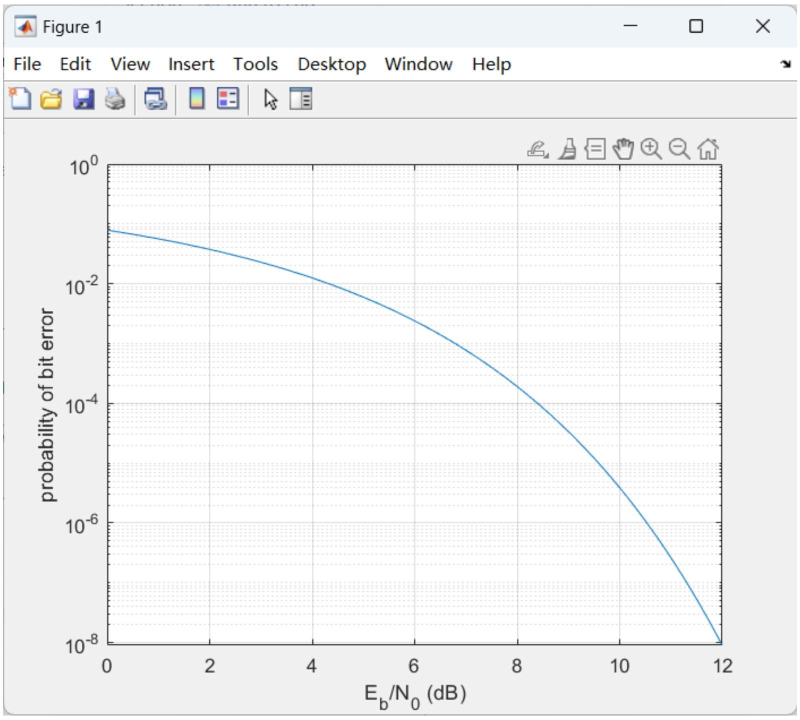

Figure 2-9 The theoretical prediction.

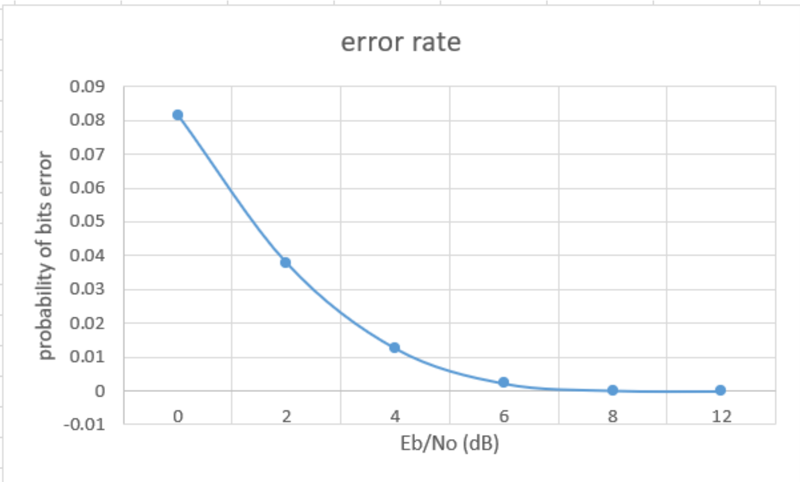

Figure 2-10 Results from MATLAB simulator.

Regarding this issue, I used Excel to help me summarise the data and generate images, Figures 2-9 and 2-10 show the graphs of the data obtained using the theory and the graphs obtained using Simulink to build the model respectively. A clear difference can be observed, and Table 2-1 shows a presentation of my results using Simulink to run the model with different parameters.

I think the main reason the difference arises is that realistically built models carry a lot of noise because it's impossible to make them disappear completely in real life.

Eb /No values	error rate
0	0.081454545
2	0.037818181
4	0.012636364
6	0.002272727
8	0.000181818
12	0

Table 2-1 Data generated by MATLAB simulator.

3. Summary

In the course of this experiment, I gained a deeper understanding of signal generation, reception, and analysis. This hands-on experience employed a broader array of analytical tools and incorporated various types of diagrams that enriched the analytical process. The comprehensive nature of this experiment not only solidified my understanding of signal behaviours but also enhanced my proficiency in utilizing the platform for complex analysis. Overall, this rigorous exercise has significantly improved my skill set in signal analysis, enabling me to approach the subject with greater confidence and expertise.